IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appl.No.: 09/494,218 Confirmation No.: 3161
Appellant: Brewer
Filed: January 28, 2000
TC/AU 2142
Examiner: Blair

Docket No.: TI-28385
Cust.No.: 23494

APPELLANT'S BRIEF

Commissioner for Patents
P.O.Box 1450
Alexandria VA 22313-1450

Sir:

The attached sheets contain the Rule 41.37 items of appellant's Appeal Brief pursuant to the Notice of Appeal filed 03/28/2007. The Director is hereby authorized to charge the fee for filing a brief in support of the appeal plus any other necessary fees to the deposit account of Texas Instruments Incorporated, account No. 20-0668.

OR

Respectfully submitted,

Carlton H. Hoel
Reg. No. 29,934
Texas Instruments Incorporated
PO Box 655474, M/S 3999
Dallas, Texas 75265
972.917.4365

Rule 41,37(c)(1)(i) Real party of interest

Texas Instruments Incorporated owns the application.

Rule 41.37(c)(1)(ii) Related appeals and interferences

There are no related dispositive appeals or interferences.

Rule 41.37(c)(1)(iii) Status of claims

Pursuant to MPEP 1205.02, for each claim in the case appellant states the status as follows:

Claim 1: rejected
Claim 2: rejected
Claim 3: rejected
Claim 4: rejected
Claim 5: rejected
Claim 6: rejected
Claim 7: rejected
Claim 8: rejected
Claim 9: rejected

Pursuant to MPEP 1205.02, appellant identifies each claim on appeal as follows

Claim 1: on appeal
Claim 2: on appeal
Claim 3: on appeal
Claim 4: on appeal
Claim 5: on appeal
Claim 6: on appeal
Claim 7: on appeal
Claim 8: on appeal
Claim 9: on appeal

Rule 41.37(c)(1)(iv) Status of amendments

There is no amendment after final rejection.

Rule 41.37(c)(1)(v) Summary of the claimed subject matter

The independent claims on appeal consist of method claim 1, method claim 2, system claim 5, method claim 7, and system claim 8.

The subject matter of claim 1 is a method for loading class files from a server to a client comprising:

(a) loading an application class onto a gateway server that preloads and preresolves said class (application page 8, lines 6-9; FIG.4, 52-53);

(b) creating a binary representation of the new portion of the preloaded and preresolved class at said gateway (application page 8, lines 10-13; FIG.4, 55); and

(c) sending only the new portion to the client (application page 8, lines 13-15; FIG.4, 56).

The subject matter of claim 2 is a method for loading Java class files from a server to a client device comprising the steps of:

(a) a gateway retrieving a Java class file (application page 8, lines 6-7; FIG.4, 52);

(b) a gateway preloading and preresolving said Java class file and creating a representation of the Java class file (application page 8, lines 7-9; FIG.4, 53);

(c) determining at the gateway the new portion of said representation of the Java class file not loaded in said client device (application page 8, lines 9-10; FIG.4, 54);

(d) creating at the gateway a binary representation of only the new portion of said representation of the Java class file (application page 8, lines 10-13; FIG.4, 55);

(e) sending said binary representation of said new portion to the client device (application page 8, lines 13-15; FIG.4, 56);

(f) loading said binary representation of said new portion into said client device (application page 8, lines 15-17; FIG.4, 57); and,

(g) copying said binary representation into the internal class structure in the interpreter of a Java virtual machine of the client device (application page 8, lines 15-17; FIG.4, 57).

The subject matter of claim 5 is a system for loading Java class files from a server to a client device comprising:

(a) a gateway coupled to said server and responsive to receipt of a Java class file for creating a c-code representation of said Java class file (application page 6, lines 18-22; FIG.3, 37, 31, 39);

(b) said gateway creating a binary representation of said c-code representation (application page 6, lines 22-26; FIG.3, 39a);

(c) a network coupled between said gateway and said client device for sending the binary representation to said client device (application page 6, lines 26-30; FIG.3, 43);

(d) a loader for loading said binary representation at said client device (application page 6, line 30; FIG.3, 47); and

(e) means for copying said binary representation into the internal class structure in an interpreter of said client device (application page 6, line 30 to page 7, line 3; FIG.3, 49).

The subject matter of claim 7 is a method for loading Java class files to an embedded client device from a server comprising the steps of:

(a) a gateway retrieving a Java class file (application page 8, lines 6-7; FIG.4, 52);

(b) a gateway preloading and preresolving the Java class file to produce a representation of the Java class file (application page 8, lines 7-9; FIG.4, 53);

(c) determining at the gateway a new portion of the representation (application page 8, lines 9-10; FIG.4, 54);

(d) creating at the gateway a binary representation of only said new portion of the preloaded and preresolved representation of the Java class file (application page 8, lines 10-13; FIG.4, 55);

(e) sending said binary representation to the embedded client device (application page 8, lines 13-15; FIG.4, 56);

(f) loading said binary representation into said embedded client device (application page 8, lines 15-17; FIG.4, 57); and

(g) copying said binary representation into the internal class structure in the interpreter of a Java virtual machine of the embedded client device (application page 8, lines 15-17; FIG.4, 57).

The subject matter of claim 8 is a system for loading Java class files from a server to an embedded client device comprising:

(a) a preloader and preresolver in a gateway (application page 6, lines 18-22; FIG.3, 37,39) coupled to said server (application page 6, lines 12-14; FIG.3, 31, 41) and responsive to receipt of a Java class file (application page 6, lines 14-18; FIG.3, 33) for preloading and preresolving a representation of said class file (application page 6, lines 18-22);

(b) said gateway creating a binary representation of said preloaded and preresolved representation of said class file (application page 6, lines 24-26; FIG.3, 39a);

(c) a wireless network coupled between said gateway and said embedded client device for sending the binary representation to said embedded device (application page 6, lines 26-30; FIG.3, 43);

(d) a loader for loading said binary representation at said embedded client device (application page 6, line 30; FIG.3, 47); and,

(e) means for copying said binary representation into the internal class structure in an interpreter of said embedded client device (application page 6, line 30 to page 7, line 3; FIG.3, 45, 49).

Rule 41.37(c)(1)(vi) Grounds of rejection to be reviewed on appeal

The grounds of rejection to be reviewed on appeal are:

1. Claim 5 was rejected under 35 USC § 102(e) as being anticipated by USP 6,295,638 to Brown et al.

2. Claims 1-3 and 6-7 were rejected under 35 USC § 103(a) as being unpatentable over USP 6,295,238 to Brown et al. in view of USP 6,389,589 to Mishra et al.

3. Claims 4 and 9 were rejected under 35 USC § 103(a) as being unpatentable over USP 6,295,238 to Brown et al. in view of USP 6,389,589 to Mishra et al. in further view of USP 6,263,360 to Arnold et al.

4. Claim 8 was rejected under 35 USC § 103(a) as being unpatentable over USP 6,295,238 to Brown et al. in view of USP 6,263,360 to Arnold et al.

Rule 41.37(c)(1)(vii) Argument

1. Claim 5 was rejected as anticipated by Brown. The Examiner cited column 7, lines 25-44 for the claimed gateway.

Claim 5: Cited Brown column 7, lines 25-44 describes creation of FCCFs (fully caffeinated class files) in a server which are delivered directly to a client without any intervening gateway; see Brown Figs.1,4. More particularly, the front-end compiler should convert java source code (column 7, lines 29-31) into an abstract syntax tree, and the back-end compiler would then generate the standard class file output (column 7, lines 31-32) which would be the directory 502 and binary 504 (bytecodes) of Fig.5; there is no gateway.

“Gateway” is standard telecommunications terminology, and a gateway at minimum connects two networks which have different protocols; e.g., Wikipedia “Gateway (telecommunications)”. Application Fig.3 illustrates this with gateway 37 on wired network 41 with server 31 and connecting to wireless network 43 with client 45. In contrast, a compiler converts source code to machine code (such as to both JVM code plus native code for FCCF in Brown) and has no suggestion of gateways.

2. Claims 1-3 and 6-7 were rejected as unpatentable over Brown in view of Mishra. The Examiner added Mishra to show sending only new portions of a file.

Claims 1-3 and 6-7: Appellant repeats the foregoing argument with regard to claim 5 that Brown neither discloses nor suggests a gateway.

3. Claims 4 and 9 were rejected as unpatentable over Brown in view of Mishra and Arnold. The Examiner added Arnold to show Java classes sent over a wireless network.

Claims 4 and 9: Appellant repeats the foregoing argument with regard to claim 5 that Brown neither discloses nor suggests a gateway.

4. Claim 8 was rejected as unpatentable over Brown in view of Arnold.

Claim 8: Appellant repeats the foregoing argument with regard to claim 5 that Brown neither discloses nor suggests a gateway.

Consequently, the claims are patentable over the references.

Rule 41.37(c)(1)(viii) Appendix

1. A method for loading class files from a server to a client comprising:

loading an application class onto a gateway server that preloads and preresolves said class;

creating a binary representation of the new portion of the preloaded and preresolved class at said gateway; and

sending only the new portion to the client.

2. A method for loading Java class files from a server to a client device comprising the steps of:

a. a gateway retrieving a Java class file;

b. a gateway preloading and preresolving said Java class file and creating a representation of the Java class file;

c. determining at the gateway the new portion of said representation of the Java class file not loaded in said client device;

d. creating at the gateway a binary representation of only the new portion of said representation of the Java class file;

e. sending said binary representation of said new portion to the client device;

f. loading said binary representation of said new portion into said client device; and,

g. copying said binary representation into the internal class structure in the interpreter of a Java virtual machine of the client device.

3. The method of Claim 2, wherein step b includes creating a c-code representation of the Java class file and step c includes determining the new portion of said c-code representation, and step d creates a binary representation of only the new portion of said c-code representation.

4. The method of Claim 2, wherein said sending step e includes sending over a wireless network.

5. A system for loading Java class files from a server to a client device comprising:

a. a gateway coupled to said server and responsive to receipt of a Java class file for creating a c-code representation of said Java class file;

b. said gateway creating a binary representation of said c-code representation;

c. a network coupled between said gateway and said client device for sending the binary representation to said client device;

d. a loader for loading said binary representation at said client device; and

e. means for copying said binary representation into the internal class structure in an interpreter of said client device.

6. The system of Claim 5, wherein said gateway includes means for determining new portions of the said c-code representation, and in step b said gateway

creates binary representations of only new portions of said c-code representations, and in step c said network sending only said new portions to said client device.

7. A method for loading Java class files to an embedded client device from a server comprising the steps of:

a. a gateway retrieving a Java class file;

b. a gateway preloading and preresolving the Java class file to produce a representation of the Java class file;

c. determining at the gateway a new portion of the representation;

d. creating at the gateway a binary representation of only said new portion of the preloaded and preresolved representation of the Java class file;

e. sending said binary representation to the embedded client device;

f. loading said binary representation into said embedded client device; and

g. copying said binary representation into the internal class structure in the interpreter of a Java virtual machine of the embedded client device.

8. A system for loading Java class files from a server to an embedded client device comprising:

a. a preloader and preresolver in a gateway coupled to said server and responsive to receipt of a Java class file for preloading and preresolving a representation of said class file;

b. said gateway creating a binary representation of said preloaded and preresolved representation of said class file;

c. a wireless network coupled between said gateway and said embedded client device for sending the binary representation to said embedded device;

d. a loader for loading said binary representation at said embedded client device; and,

e. means for copying said binary representation into the internal class structure in an interpreter of said embedded client device.

9. The system of Claim 8, wherein said gateway includes means for determining new portions of said preloaded and preresolved representations of the class and sending only said new portions to said embedded client device.

Rule 41.37(c)(1)(ix) Evidence appendix

none

Rule 41.37(c)(1)(x) Related proceedings appendix

none